AUDIT REPORT

2023

ANALYTIX AUDIT

Security Assessment

Milady 2.0 Token

July 3, 2023

Audit Status: Pass

Audit Edition: Advance

Risk Analysis

Classifications of Manual Risk Results

Classification	Description
Critical	Danger or Potential Problems.
Major	Be Careful
Minor	Pass, Not-Detected or Safe Item.
Informational	Function Detected

Manual Code Review Risk Results

Contract Priviledge	Description
Can mint?	Pass
Edit taxes over 25%?	Pass
Max Tx?	Pass
Max Wallet?	Pass
Has to enable trading?	Trading is already enabled.
Modify Tax	Pass
Can blacklist?	Pass
Is Honeypot?	Liquidity has not been added
Trading Cooldown	Not Detected
Can Pause Trade?	Pass
Pause Transfer?	Not Detected

Contract Priviledge	Description
Is Proxy??	Not Detected
Is Anti Whale?	Not Detected
Is Anti Bot?	Not Detected
Is Blacklist?	Not Detected
Blacklist Check	Pass
is Whitelist?	Not Detected
Buy Tax	0
Sell Tax	0
Can Take Ownership?	Not Detected
Hidden Owner?	Not Detected
Owner	0x0000000000000000000000000000000000000000
Self Destruct?	Not Detected
Other?	Not Detected
Other?	Not Detected
Holders	1
Auditor Confidence	Medium

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Table of Contents

Assessment Summary

This report has been prepared for Milady 2.0 Token on the Ethereum Main Network network. AnalytixAudit provides both client-centered and user-centered examination of the smart contracts and their current status when applicable. This report represents the security assessment made to find issues and vulnerabilities on the source code along with the current liquidity and token holder statistics of the protocol.

A comprehensive examination has been performed, utilizing Cross Referencing, Static Analysis, In-House Security Tools, and line-by-line Manual Review.

The auditing process pays special attention to the following considerations:

- Testing the smart contracts against both common and uncommon attack vectors.
- Inspecting liquidity and holders statistics to inform the current status to both users and client when applicable.
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Verifying contract functions that allow trusted and/or untrusted actors to mint, lock, pause, and transfer assets.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders
- Thorough line-by-line manual review of the entire codebase by industry experts.

Project Overview

Token Summary

Parameter	Result
Address	0x7EaCFA5147cC1EBb49433d6FC41468b883E46344
Name	Milady 2.0
Token Tracker	Milady 2.0 (LADYS2.0)
Decimals	18
Supply	888,000,888,000,888
Platform	Ethereum Main Network
compiler	v0.8.7+commit.e28d00a7
Contract Name	Ladys2
Optimization	No
LicenseType	MIT
Language	Solidity
Codebase	https://etherscan.io//address/0x7EaCFA5147cC1EBb49433d6FC41468b883E46344#code
Payment Tx	Corporate

Project Overview

Simulation Summary

Parameter	Result
Transfer From Owner	Pass
Transfer From Holder	Pass
Add Liquidity	Pass
RemoveLiquidity	Pass
Buy from Owner	Pass
Buy from Holder	Pass
Sale from Owner	Pass
Sale from Holder	Pass
Remove Liquidity	Pass
SwapAndLiquify	Pass
SwapAndSale w/Fee	Pass
SwapAndSale TX	
SwapAndSaleNoFee	Pass
SwapAndSale No/Fee TX	
ExcludeFromFees	Pass
LaunchPad	PinkSale

Parameter	Result
Pool Creation	Pass
Pool Creation TX	
Pool Finalize	Pass
Pool Finalize TX	
Enable	Pass

The following quick summary it's added to the project overview; however, there are more details about the audit and its results. Please read every detail.

Main Contract Assessed
Contract Name

Name	Contract	Live
Milady 2.0	0x7EaCFA5147cC1EBb49433d6FC41468b883E46344	Yes

TestNet Contract was Not Assessed

Solidity Code Provided

SolID	File Sha-1	FileName
LADYS2.0	9eb45379dccf041954c13960d4e559933834c11	LADYS2.0.sol
LADYS2.0		
LADYS2.0		
LADYS2.0		

KYC Information

The Project Owners of Milady 2.0 is not KYC.

KYC Information Notes:

Auditor Notes: No info founde

Project Owner Notes:

Smart Contract Vulnerability Checks

The Smart Contract Weakness Classification Registry (SWC Registry) is an implementation of the weakness classification scheme proposed in EIP-1470. It is loosely aligned to the terminologies and structure used in the Common Weakness Enumeration (CWE) while overlaying a wide range of weakness variants that are specific to smart contracts.

ID	Severity	Name	File	location
SWC-100	Pass	Function Default Visibility	LADYS2.0.sol	L: 0 C: 0
SWC-101	Pass	Integer Overflow and Underflow.	LADYS2.0.sol	L: 0 C: 0
SWC-102	Pass	Outdated Compiler Version file.	LADYS2.0.sol	L: 0 C: 0
SWC-103	Low	A floating pragma is set.	LADYS2.0.sol	L: 6 C: 0
SWC-104	Pass	Unchecked Call Return Value.	LADYS2.0.sol	L: 0 C: 0
SWC-105	Pass	Unprotected Ether Withdrawal.	LADYS2.0.sol	L: 0 C: 0
SWC-106	Pass	Unprotected SELFDESTRUCT Instruction	LADYS2.0.sol	L: 0 C: 0
SWC-107	Pass	Read of persistent state following external call.	LADYS2.0.sol	L: 0 C: 0
SWC-108	Pass	State variable visibility is not set..	LADYS2.0.sol	L: 0 C: 0
SWC-109	Pass	Uninitialized Storage Pointer.	LADYS2.0.sol	L: 0 C: 0
SWC-110	Pass	Assert Violation.	LADYS2.0.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-111	Pass	Use of Deprecated Solidity Functions.	LADYS2.0.sol	L: 0 C: 0
SWC-112	Pass	Delegate Call to Untrusted Callee.	LADYS2.0.sol	L: 0 C: 0
SWC-113	Pass	Multiple calls are executed in the same transaction.	LADYS2.0.sol	L: 0 C: 0
SWC-114	Pass	Transaction Order Dependence.	LADYS2.0.sol	L: 0 C: 0
SWC-115	Pass	Authorization through tx.origin.	LADYS2.0.sol	L: 0 C: 0
SWC-116	Pass	A control flow decision is made based on The block.timestamp environment variable.	LADYS2.0.sol	L: 0 C: 0
SWC-117	Pass	Signature Malleability.	LADYS2.0.sol	L: 0 C: 0
SWC-118	Pass	Incorrect Constructor Name.	LADYS2.0.sol	L: 0 C: 0
SWC-119	Pass	Shadowing State Variables.	LADYS2.0.sol	L: 0 C: 0
SWC-120	Pass	Potential use of block.number as source of randonmness.	LADYS2.0.sol	L: 0 C: 0
SWC-121	Pass	Missing Protection against Signature Replay Attacks.	LADYS2.0.sol	L: 0 C: 0
SWC-122	Pass	Lack of Proper Signature Verification.	LADYS2.0.sol	L: 0 C: 0
SWC-123	Pass	Requirement Violation.	LADYS2.0.sol	L: 0 C: 0
SWC-124	Pass	Write to Arbitrary Storage Location.	LADYS2.0.sol	L: 0 C: 0
SWC-125	Pass	Incorrect Inheritance Order.	LADYS2.0.sol	L: 0 C: 0

ID	Severity	Name	File	location
SWC-126	Pass	Insufficient Gas Griefing.	LADYS2.0.sol	L: 0 C: 0
SWC-127	Pass	Arbitrary Jump with Function Type Variable.	LADYS2.0.sol	L: 0 C: 0
SWC-128	Pass	DoS With Block Gas Limit.	LADYS2.0.sol	L: 0 C: 0
SWC-129	Pass	Typographical Error.	LADYS2.0.sol	L: 0 C: 0
SWC-130	Pass	Right-To-Left-Override control character (U +202E).	LADYS2.0.sol	L: 0 C: 0
SWC-131	Pass	Presence of unused variables.	LADYS2.0.sol	L: 0 C: 0
SWC-132	Pass	Unexpected Ether balance.	LADYS2.0.sol	L: 0 C: 0
SWC-133	Pass	Hash Collisions with Multiple Variable Length Arguments.	LADYS2.0.sol	L: 0 C: 0
SWC-134	Pass	Message call with hardcoded gas amount.	LADYS2.0.sol	L: 0 C: 0
SWC-135	Pass	Code With No Effects (Irrelevant/Dead Code).	LADYS2.0.sol	L: 0 C: 0
SWC-136	Pass	Unencrypted Private Data On-Chain.	LADYS2.0.sol	L: 0 C: 0

We scan the contract for additional security issues using MYTHX and industry-standard security scanning tools.

Smart Contract Vulnerability Details

SWC-103 - Floating Pragma.

CWE-664: Improper Control of a Resource Through its Lifetime.

References:

Description:

Contracts should be deployed with the same compiler version and flags that they have been tested with thoroughly. Locking the pragma helps to ensure that contracts do not accidentally get deployed using, for example, an outdated compiler version that might introduce bugs that affect the contract system negatively.

Remediation:

Lock the pragma version and also consider known bugs (https://github.com/ethereum/solidity/releases) for the compiler version that is chosen.

Pragma statements can be allowed to float when a contract is intended for consumption by other developers, as in the case with contracts in a library or EthPM package. Otherwise, the developer would need to manually update the pragma in order to compile locally.

References:

Ethereum Smart Contract Best Practices - Lock pragmas to specific compiler version.

Inheritance

The contract for Milady 2.0 has the following inheritance structure.

The Project has a Total Supply of 888,000,888,000,888

Smart Contract Advance Checks

ID	Severity	Name	Result	Status
LADYS2.0-01	Minor	Potential Sandwich Attacks.	Pass	Not-Found
LADYS2.0-02	Minor	Function Visibility Optimization	Pass	Not-Found
LADYS2.0-03	Minor	Lack of Input Validation.	Pass	Not-Found
LADYS2.0-04	Major	Centralized Risk In addLiquidity.	Pass	Not-Found
LADYS2.0-05	Minor	Missing Event Emission.	Pass	Not-Found
LADYS2.0-06	Minor	Conformance with Solidity Naming Conventions.	Pass	Not-Found
LADYS2.0-07	Minor	State Variables could be Declared Constant.	Pass	Not-Found
LADYS2.0-08	Minor	Dead Code Elimination.	Pass	Not-Found
LADYS2.0-09	Major	Third Party Dependencies.	Pass	Not-Found
LADYS2.0-10	Major	Initial Token Distribution.	Pass	Not-Found
LADYS2.0-11	Major	Complexity on the tax calculations.	Pass	Not-Found
LADYS2.0-12	Major	Centralization Risks In The X Role	Pass	Not-Found
LADYS2.0-13	Informational	Extra Gas Cost For User..	Pass	Not-Found

ID	Severity	Name	Result	Status
LADYS2.0-14	Medium	Unnecessary Use Of SafeMath	Pass	Not-Found
LADYS2.0-15	Medium	Symbol Length Limitation due to Solidity Naming Standards.	Pass	Not-Found
LADYS2.0-16	Medium	Invalid collection of Taxes during Transfer.	Pass	Not-Found
LADYS2.0-17	Informational	Conformance to numeric notation best practice.	Pass	Not-Found
LADYS2.0-18	Informational	Enable Trade and Exclude Exist to create a whitelist.	Pass	Not-found

Technical Findings Summary

Classification of Risk

Severity	Description
Critical	Risks are those that impact the safe functioning of a platform and must be addressed before launch. Users should not invest in any project with outstanding critical risks.
Major	Risks can include centralization issues and logical errors. Under specific circumstances, these major risks can lead to loss of funds and/or control of the project.
Medium	Risks may not pose a direct risk to users' funds, but they can affect the overall functioning of a platform
Minor	Risks can be any of the above but on a smaller scale. They generally do not compromise the overall integrity of the Project, but they may be less efficient than other solutions.
Informational	Errors are often recommended to improve the code's style or certain operations to fall within industry best practices. They usually do not affect the overall functioning of the code.

Findings

Severity	Found	Pending	Resolved
Critical	0	0	0
Major	0	0	0
Medium	0	0	0
Minor	1	1	0
Informational	0	0	0
Total	1	1	0

ANALYTIX AUDIT

Social Media Checks

Social Media	URL	Result
Twitter	https://twitter.com/_Milady2	Pass
Other		Fail
Website	https://milady20.com/	Pass
Telegram	https://t.me/LadysMilady2	Pass

We recommend to have 3 or more social media sources including a completed working websites.

Social Media Information Notes:

Auditor Notes: undefined

Project Owner Notes:

Assessment Results

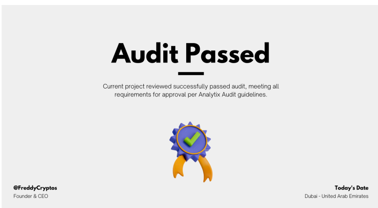
Score Results

Review	Score
Overall Score	89/100
Auditor Score	90/100

Review by Section	Score
Manual Scan Score	36/53
SWC Scan Score	36 /37
Advance Check Score	17 /19

The Following Score System Has been Added to this page to help understand the value of the audit, the maximun score is 100, however to attain that value the project most pass and provide all the data needed for the assessment. Our Passing Score has been changed to 80 Points, if a project does not attain 80% is an automatic failure. Read our notes and final assessment below.

Audit Passed

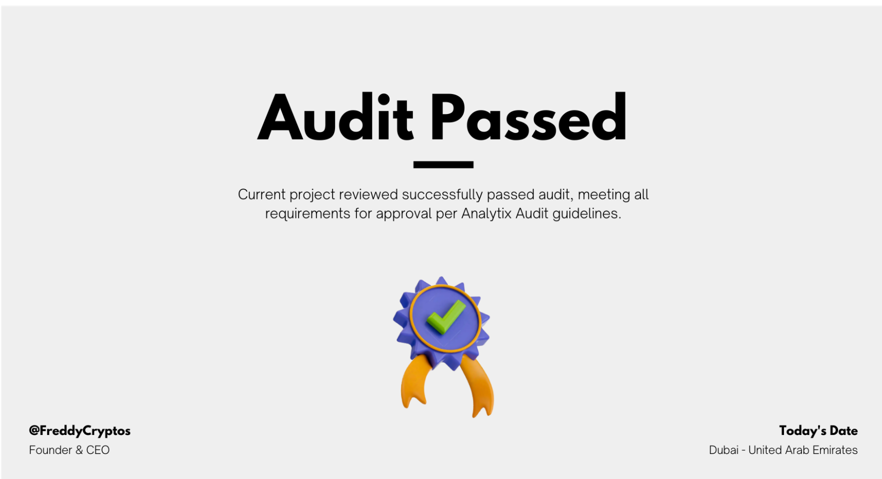


Assessment Results

Important Notes:

- No High-Risk Issues or vulnerabilities were found.
- Always DYOR on the project itself.

Auditor Score =90
Audit Passed



Appendix

Finding Categories

Centralization / Privilege

Centralization / Privilege findings refer to either feature logic or implementation of components that actagainst the nature of decentralization, such as explicit ownership or specialized access roles incombination with a mechanism to relocate funds.

Gas Optimization

Gas Optimization findings do not affect the functionality of the code but generate different, more optimalEVM opcodes resulting in a reduction on the total gas cost of a transaction.

Logical Issue

Logical Issue findings detail a fault in the logic of the linked code, such as an incorrect notion on howblock.timestamp works.

Control Flow

Control Flow findings concern the access control imposed on functions, such as owner-only functionsbeing invoke-able by anyone under certain circumstances.

Volatile Code

Volatile Code findings refer to segments of code that behave unexpectedly on certain edge cases that mayresult in a vulnerability.

Coding Style

Coding Style findings usually do not affect the generated byte-code but rather comment on how to makethe codebase more legible and, as a result, easily maintainable.

Inconsistency

Inconsistency findings refer to functions that should seemingly behave similarly yet contain different code,such as a constructor assignment imposing different require statements on the input variables than a setterfunction.

Coding Best Practices

ERC 20 Conding Standards are a set of rules that each developer should follow to ensure the code meet a set of creterias and is readable by all the developers.

Disclaimer

AnalytixAudit has conducted an independent security assessment to verify the integrity of and highlight any vulnerabilities or errors, intentional or unintentional, that may be present in the reviewed code for the scope of this assessment. This report does not constitute agreement, acceptance, or advocation for the Project, and users relying on this report should not consider this as having any merit for financial advice in any shape, form, or nature. The contracts audited do not account for any economic developments that the Project in question may pursue, and the veracity of the findings thus presented in this report relate solely to the proficiency, competence, aptitude, and discretion of our independent auditors, who make no guarantees nor assurance that the contracts are entirely free of exploits, bugs, vulnerabilities or deprecation of technologies.

All information provided in this report does not constitute financial or investment advice, nor should it be used to signal that any persons reading this report should invest their funds without sufficient individual due diligence, regardless of the findings presented. Information is provided 'as is, and AnalytixAudit is under no covenant to audited completeness, accuracy, or solidity of the contracts. In no event will AnalytixAudit or its partners, employees, agents, or parties related to the provision of this audit report be liable to any parties for, or lack thereof, decisions or actions with regards to the information provided in this audit report.

The assessment services provided by AnalytixAudit are subject to dependencies and are under continuing development. You agree that your access or use, including but not limited to any services, reports, and materials, will be at your sole risk on an as-is, where-is, and as-available basis. Cryptographic tokens are emergent technologies with high levels of technical risk and uncertainty. The assessment reports could include false positives, negatives, and unpredictable results. The services may access, and depend upon, multiple layers of third parties.